**Abstract** - The ability to separate the essential features of an object (object) from the secondary, to see the idea to be implemented.

**Requirements Analysis** - Display functions of the system and its limitations in the domain model.

the software architecture of the system - the system structure in terms of subsystems (components) and the interfaces between them, mapping the problem of decomposition rules.

**White-box method** - study of the internal structure of the system in order to detect errors in all of these ways of its transmission and flow control.

**Waterfall (cascade) model -** a scheme of works in which each work is done only once and in the order specified in the life-cycle model.

**Guaranteed quality software** - actions at each stage of the life cycle of the verification and confirmation of quality achieved, respectively, standards and procedures.

**Defect -** this error event in the system, resulting from an incorrect description of the requirements of the specifications, design specifications, documentation, etc.

**Chart** - a graphical representation scenarios using the system classes, conditions, events, etc

ynamic testing - the program to detect errors, determine their causes and remedies.

**The life cycle of the system -** the scheme of work on the design of the system, starting from the moment the decision on the need for its creation to the moment of its complete withdrawal from service.

**System Task** - method (technology) to achieve system goals with specific numeric (including timing) characteristics.

**Simulation** - simulation of the behavior of the system in various aspects and in various external and internal conditions of the analysis of dynamic performance and resource allocation.

Engineering - the application of scientific results and programming tasks of management disciplines in order to benefit from the properties of the products, methods and implementation of the relationship.

**Quality Engineering** - Product management process providing software quality properties (reliability, fault tolerance, etc.).

**Engineering requirements** - collection, analysis, execution conditions and restrictions on the development of the system in the form of a specification agreed by both the customer and the performer.

Incremental model- it works in parallel with the continuous analysis of the results and the adjustment of the previous work phases. The project with this approach in each phase of development goes repeating cycle PDCA: Planning - Implementation - Testing - Evaluation (English plan-do-check-act cycle.).

**Code inspection** - formal verification description used types and data structures in the design of the system on their eligibility.

**Information model -** model of the system, which displays the data structure and relationships with objects that were used.

**Information system** - a system that performs the collection, processing, preservation and production of information using automated processors and people.

**Information support** - a set of tools to provide information to users about the contents and the conditions of its application.

**Quality software** - a set of properties that determine the suitability of the software to meet customer requirements.

**Methods of targeting data** (data-oriented methods- base data structures manipulated to create software.

**Life cycle model -** the template for a sequence of works on the development processes of a certain type of software.

**Process Model** - defined sequence of steps, accompanied by changes in the state of the program object.

The reliability of the software system - it is the system's ability to maintain its properties (reliability, stability, etc.). In the process of transforming raw data into results for a certain period of time under certain conditions.

**Debugging - the program c**hecks for errors in its specification and their removal without introducing new ones.

**Disclaimer** - the transition program of the operating state when not in connection with the detected errors or defects in it.

**Error -** shortcomings in the program statement or in the process of its development, which lead to misinterpretation of the initial information and the wrong decision.

**Object of the essence** - long-lived objects that respond to the real objects and the domain of the world preserve their state after the job according to the scenario.

**The test plan** - description of strategy, resources and schedule testing of individual components and the system as a whole.

**Behavior domain** - the domain elements transition from state to state in time.

**Reuse** - use as a finished portion of any formal knowledge gained in the implementation of software systems.

**Applied system** - a product of software engineering for performing end-user specific tasks.

Principles - the basic concepts underlying the entire programming basis.

**Appendix** - field of application in which the principles, methods and practices find their best expression.

**Software engineering** - a system of methods, tools and planning discipline, development, operation and maintenance of software, capable to mass reproduction.

**The acquisition process** - actions that trigger a certain cycle analysis to determine the buyer of a software system or service.

**The development process** - action Developer's engineering requirements, design, coding and testing of software.

**The process of surrender** - actions developed product transfer to the buyer.

operation Process - maintenance activities of the system user.

**Process support** - actions to address system problems, support systems to date to perform the functions of the system, control system modifications or withdrawal from use.

**Design** - conversion requirements in order to design solutions and system architecture.

**Conceptual design** - refinement of understanding and agreement details the system requirements.

Architectural design - defining the structural features of the system under construction.

**Design engineering** - mapping of the system and development environment requirements by defining all of the structural elements and their compositions.

**Detailed design** - defining the implementation details of functions for a given environment and the relationships between these components of the system.

**Prototyping software** (from the English prototyping.) - Stage of software development (software), the process of creating software prototype - layout (roughing, trial) program - usually for the purpose of fitness checks proposed to apply the concepts of architectural and / or technological solutions, as well as for the presentation of the program to the customer in the early stages of the development process.

**Specification** - description of the algorithm, rules, constraints of existing facilities based on standards, quality criteria and others.

**Spiral model of life cycle** - model of the processes of system development, with the ability to return to any previous process with the aim of processing elements to make the product.

**Event** - a phenomenon that provokes a change of certain state and a transition to another state in the system.

**Condition (domain system object and the like**), - fixing of certain properties or a certain time interval.

**Static testing** - Analysis and review of specifications on components are correctly represented without their execution on the computer.

**Maintenance** - perform tasks implemented in the system, work on the amended after it is transmitted to the users manual.

**The structure of the system** - a set of elements and the relationships between them.

**Structural methods** (structured methods). With this approach, the system is built from a functional point of view, starting with the high-level understanding of the behavior of the system with the gradual refinement of the low-level details

**Essence (Entity**) - real or imaginary object, which has a significant importance for the subject area, information about which is to be stored.

**Scenario** - a specific sequence of actions that illustrates the behavior and performance of an instance of a precedent.

**The test** - some program designed to validate its work and identify it erroneous situations.

**Test data** - a set of data, which are prepared on the basis of a program or specification documents, to test the software system.

**Testing** - semantic method of debugging (test) program, which is to perform a series of different control test suites and validation of the results obtained with the quadrupeds in advance.

**The requirement** - agreement or contract between the customer and the system with respect to the properties of its functions, the conditions of work in a given environment.

**Function -** action content, the implementation of which is assigned to the element of the system with the specified requirements, conditions and limitations.

**Functional requirements** - these are the conditions and limitations for the purposes of the system functions and how they perform on the computer.

**Black box method** - testing the functions implemented by verifying compliance with the actual behavior of functions with the expected behavior, based on the specification of requirements.

**Evolutionary model lifecycle** - requirements for IP set and partially refined in each successive intermediate structure of the system unit.

**Heuristic methods (Heuristic Methods)** - a sequence of instructions or data processing procedures performed in order to find more efficient and new design solutions

**Occurrence** - abstract relationship between a parameterized class-template (template) and the real class cherezopredelenie template parameters.

**Operation** - action to implement the finished software system.